Appl. No. 10/735,482
Amdt. dated June 16, 2008
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group 2193

PATENT

REMARKS/ARGUMENTS

Status of the Claims

Claims 1 and 4-50 are currently pending in the application. Claims 1, 36, 38, 39, 44, and 46 have been amended. No new matter has been added by the amendments. Claim 44 has been cancelled. No claims have been added. Therefore, claims 1, 4-43, and 45-50 are present for examination. Claims 1, 36, 38, 39, 45, and 46 are independent claims.

Prior to entry of this amendment, the application included claims 1 and 4-50. A final office action mailed April 16, 2008, has rejected claims 1, 4-18, and 20-50 under 35 U.S.C. § 103(a) as being as being unpatentable over "Java[tm] 2 Platform, Standard Edition v1.2.2 API Specification ("**Java SE**") in view of "Specification-based Testing for GUI-based Applications" by Chen et al. ("**Chen**").

35 U.S.C. §103 Rejection, Java SE in view of Chen et al.

Claims 1, 4-18, and 20-50 have been rejected under 35 U.S.C. 103(a) as being unpatentable over "Java[tm] 2 Platform, Standard Edition v1.2.2 API Specification ("**Java SE**") in view of "Specification-based Testing for GUI-based Applications" by Chen et al. ("**Chen**").

Java SE generally discloses an API specification for the Java 2 Platform, Standard Edition, version 1.2.2. This API specification includes a variety of Java packages, classes, methods, etc. Specifically, Java SE discloses a class "AccessibleContext" which represents the minimum information all accessible objects return. The information includes the accessible name, description, role, and state of the object. The AccessibleContext class includes a getAccessibleAt(Point p) method which returns an accessible child contained at the coordinate Point. AccessibleContext further includes the methods getAccessibleComponent(), getAccessibleRole() and getAccessibleAction(), which are used to manipulate an AccessibleContext object. (see Java SE at "class AccessibleContext" and "interface AccessibleComponet"). Ultimately, the AccessibleContext class allows for the retrieval of pointer data generated by an input device (e.g., a mouse).

Appl. No. 10/735,482
Amdt. dated June 16, 2008
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group 2193

PATENT

Chen generally discloses specification-based testing for GUI-based applications; such "GUI-based applications are based on the Capture/Replay technique. With this technique, in the first run of an application, a Capture/Replay tool is used to record all interesting events in a test script written in certain script languages. For input, [the test] records every point and click applied to the GUI application in a test script that is in C-like Test Script Language (TSL)." (Chen at page 207, section 2).

In contrast, claim 1 and similarly claims 36, 38, 39, and 45-46, as amended, recite the operation of "*recording, in real time and independent of any of the plurality of operating systems*, the identified accessibility context, wherein the recording of the identified accessibility context is a*n operating system independent process*." (Emphasis provided.) Applicants submit that both Java SE and Chen fail to teach or suggest such a limitation. In fact, nowhere in Java SE or Chen are Applicants able to find any disclosure of either recording an accessibility context in real time, or that the recording is operating system independent. Furthermore, Chen, for example discloses that its testing tool is for X Windows, which is a GUI specific to Linux or UNIX operating systems. (Chen at page 207, section 2). Additionally, the test scripts in Chen are described as using C-like Test Script Language (TSL), which, as known by one of ordinary skill in the art, is an operating system specific test script. (*Id.* at page 207, section 2). Hence, for at least these reasons, Applicants submit that Java SE and Chen, individually, or in combination, fail to teach of suggest the operation of "recording, in real time and independent of any of the plurality of operating systems, the identified accessibility context, wherein the recording of the identified accessibility context is an operating system independent process" as recited by claim 1. Therefore, Applicants submit that claims 1, 36, 38, 39, and 45-46 are believed to be allowable over the combination of Java SE and Chen.

Furthermore, dependent claims 4-12, 20-27, and 40-42 depend from claims 1 and 39 and therefore are believed to be allowable over the combination of Java SE and Chen at least by virtue of their dependence from allowable base claims.

Appl. No. 10/735,482
Amdt. dated June 16, 2008
Amendment under 37 CFR 1.116 Expedited Procedure
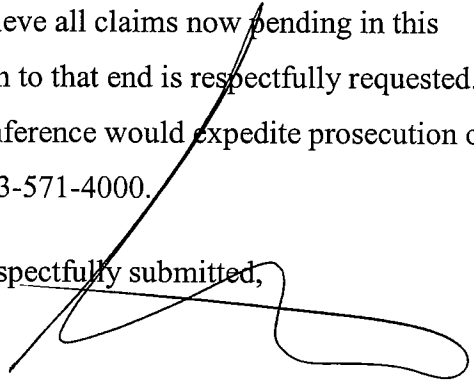Examining Group 2193

PATENT

Official Notice

Official Notice is taken that XML was widely known and used at the time of the invention, and it would have been obvious to use XML to store the records because it would have provided a system independent method of storing the data. Applicants respectfully traverse this taking of Official Notice.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-571-4000.

Respectfully submitted,

Charles W. Gray
Reg. No. 61,345

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000
Fax: 415-576-0300
CWG:slb
61358894 v1